I Erläuterungen

Voraussetzungen gemäß KCBG und Abiturerlassen BG jeweils in der für den Abiturjahrgang geltenden Fassung

Standardbezug
Die nachfolgend ausgewiesenen Kompetenzbereiche sind für die Bearbeitung der jeweiligen Aufgabe besonders bedeutsam. Darüber hinaus können weitere, hier nicht ausgewiesene Kompetenzbereiche für die Bearbeitung der Aufgabe nachrangig bedeutsam sein, zumal die Kompetenzbereiche in engem Bezug zueinander stehen. Die Operationalisierung des Bezugs zu den Kompetenzbereichen des Standardbezugs erfolgt in Abschnitt II.

Aufgabe	Kompetenzbereiche				
	K1	K2	K3	K4	K5
1.1			X		
1.2			X		
1.3				X	
2.1	X				
2.2				X	
2.3			X		
2.4			X		
2.5.1		X			
2.5.2			X		
3.1					X
3.2			X		
3.3				X	
3.4		X			
3.5.1		X			X
3.5.2			X		

Inhaltlicher Bezug
Die nachfolgend ausgewiesenen Themenfelder sind die wesentliche inhaltliche Grundlage für die vorliegenden Aufgaben. Darüber hinaus können weitere, hier nicht explizit ausgewiesene Themenfelder für die Bearbeitung nachrangig bedeutsam sein.
Q1: Objektorientierte Softwareentwicklung
Q2: Digitale Steuerungstechnik
Q3: Prozessautomatisierung
verbindliche Themenfelder: Objektmodellierung (Q1.1), Implementierung von Klassen und ihren Beziehungen (Q1.2), Synthese statischer und sequentieller Logikschaltungen (Q2.1), Mikrocontroller (Q2.2), Einführung in die Prozessautomatisierung (Q3.1)

II Lösungshinweise

In den nachfolgenden Lösungshinweisen sind alle wesentlichen Gesichtspunkte, die bei der Bearbeitung der einzelnen Aufgaben zu berücksichtigen sind, konkret genannt und diejenigen Lösungswege aufgezeigt, welche die Prüflinge erfahrungsgemäß einschlagen werden. Selbstverständlich sind jedoch Lösungswege, die von den vorgegebenen abweichen, aber als gleichwertig betrachtet werden können, ebenso zu akzeptieren.

Nicht für den Prüfling bestimmt

Aufg.	erwartete Leistungen	BE I	BE II	BE III
1.1	entwickeln, bestimmen			
	entwickeln		4	
	bestimmen	5		
1.2	beschreiben, erklären Zwischen den Klassen `User` und `Verwaltung` besteht die Assoziation `verwaltet`, welche in Richtung Verwaltung nicht navigierbar ist. Ein User gehört dabei zu einer Verwaltung, während eine Verwaltung keinen oder mehrere User haben kann. Zwischen der Klasse `Verwaltung` und `Pelletofen` besteht die Assoziation `steuert`. Eine Verwaltung kann keinen oder mehrere Pelletöfen steuern, während ein Pelletofen immer von einer Verwaltung gesteuert wird. Zwischen den Klassen `Pelletofen` und `Bauteile` besteht die Ganzes-Teil-Beziehung (Komposition) `besteht aus`, d.h. wenn es keine Instanz der Klasse `Pelletofen` gibt, existieren auch keine `Bauteile`-Objekte, bzw. Objekte der erbenden Klassen. Ein Pelletofen kann keine oder mehrere Bauteile haben, während ein Bauteil immer einem Pelletofen zugeordnet ist. Die abstrakte Klasse `Bauteile` (von ihr können keine Instanzen erstellt werden) hat das Klassenattribut (statisch, auf das direkt zugegriffen werden kann) public `anzahl` vom Typ int. Auf die protected-Attribute `name` (String) und `laufendeNr` (int) können die erbenden Klassen `Rauchgebläse`, `Zündeinrichtung` und `Thermostat` zugreifen. Alle drei erbenden Klassen haben differenzierte private Attribute vom Typ int für die jeweilige Nutzung, z.B. `tempIst` der Klasse `Zündeinrichtung`.			
	beschreiben	4		
	erklären			5

<table>
<tr><th rowspan="2">Aufg.</th><th rowspan="2">erwartete Leistungen</th><th colspan="3">BE</th></tr>
<tr><th>I</th><th>II</th><th>III</th></tr>
<tr><td>1.3</td><td>implementieren

<u>Klasse Pelletofen:</u>


```
import java.util.ArrayList;

public class Pelletofen {

	private String produktbezeichnung;
	private int seriennummer;
	private int pelletfuellstand;
	private boolean istAn;
	private String modus;
	private ArrayList<Bauteil> bauteilListe;

	public Pelletofen(String produktbezeichnung, int
seriennummer, int pelletfuellstand, boolean istAn, String
modus) {
		this.produktbezeichnung = produktbezeichnung;
		this.seriennummer = seriennummer;
		this.pelletfuellstand = pelletfuellstand;
		this.istAn = istAn;

		// Überprüfung, ob der Modus einer der
beschriebenen Modi ist
		modusEinstellen(modus);

		// Erzeugen der ArrayLists
		bauteilListe = new ArrayList<Bauteil>();

		// Umsetzung der Komposition
		rauchgeblaeseHinzufuegen("RB1", 0, 150);
	}

	public void modusEinstellen(String m) {
		if (m.equals("Brennvorgang starten") ||
m.equals("Brennvorgang stoppen") || m.equals("Normaler
Heizbetrieb")) {
			this.modus = m;
		} else {
			System.out.println("Fehler: Modus nicht
korrekt!");
			this.modus = null;
		}
	}


	public void rauchgeblaeseHinzufuegen(String n, int
ui, int us) {
		Rauchgeblaese rb = new Rauchgeblaese(n, ui,
us);
		bauteilListe.add(rb);
	}
}
```

</td><td></td><td>7</td><td>3</td></tr>
</table>

Aufg.	erwartete Leistungen	BE I	BE II	BE III

	<u>Klasse Bauteil:</u>
	`public abstract class Bauteil {`
	`    protected String name;`
	`    protected int laufendeNr;`
	`    public static int anzahl;`
	`    public Bauteil() {`
	`        Bauteil.anzahl += 1;`
	`    }`
	`}`
	<u>Klasse Rauchgebläse:</u>
	`public class Rauchgeblaese extends Bauteil {`
	`    public int umdrehungenIst;`
	`    public int umdrehungenSoll;`
	`    public Rauchgeblaese(String n, int ui, int us) {`
	`        this.umdrehungenIst = ui;`
	`        this.umdrehungenSoll = us;`
	`        super.name = n;`
	`    }`
	`}`			

	Summe 28	**9**	**11**	**8**

Nicht für den Prüfling bestimmt

Nicht für den Prüfling bestimmt

<table>
<tr><th rowspan="2">Aufg.</th><th rowspan="2">erwartete Leistungen</th><th colspan="3">BE</th></tr>
<tr><th>I</th><th>II</th><th>III</th></tr>
<tr><td>2.1</td><td>beschreiben

Ein Gleichstrommotor dient der Umwandlung von elektrischer Energie in mechanische Energie, mit der mechanische Arbeit verrichtet wird. Dabei wird eine Drehbewegung erzeugt, die man zum Antrieb von Geräten und Anlagen verwendet. Genutzt wird das Prinzip, dass auf einen stromdurchflossenen Leiter im Magnetfeld eine Kraft wirkt. Durch einen Feldmagneten (Dauer- oder Elektromagnet) wird ein magnetisches Feld aufgebaut. In diesem Feld ist ein Elektromagnet drehbar gelagert - der Anker. Über Kohlebürsten als Schleifkontakte wird der Anker an eine Stromquelle angeschlossen. Durch den Stromfluss im Anker wird dieser magnetisch, wodurch Kräfte zwischen Feldmagneten und Anker auftreten. Gleiche Magnetpole stoßen sich ab, ungleiche Magnetpole ziehen sich an. Diese abstoßenden und anziehenden Kräfte zwischen den Magnetpolen führen zu einer Drehbewegung des Ankers. Wenn sich die ungleichen Magnetpole des Feldmagneten und des Ankers direkt gegenüberstehen, muss das Magnetfeld des Ankers umgepolt werden, damit sich die Drehbewegung fortsetzt. Dies geschieht durch den Kollektor (Polwender). Durch die Trägheit der Drehbewegung rotiert der Anker über den Totpunkt hinweg. Die Stromrichtung am Anker wird umgekehrt und die Drehbewegung fortgesetzt.</td><td>4</td><td></td><td></td></tr>
<tr><td>2.2</td><td>ergänzen

(Tabelle siehe unten)</td><td>1</td><td>3</td><td></td></tr>
</table>

Index	Heiz-vorgang gestartet	Error	Temperatur Brennraum > 30 ° C	Zündkerze ein	Rauchgas-gebläse ein	Förder-schnecke ein
	d	**c**	**b**	**a**	**x**	**y**
0	0	0	0	0	**0**	**0**
1	0	0	0	1	**1**	**0**
2	0	0	1	0	**1**	**X**
3	0	0	1	1	**1**	**X**
4	0	1	0	0	**0**	**0**
5	0	1	0	1	**0**	**0**
6	0	1	1	0	**0**	**0**
7	0	1	1	1	**0**	**0**
8	1	0	0	0	**1**	**1**
9	1	0	0	1	**1**	**1**
10	1	0	1	0	**1**	**1**
11	1	0	1	1	**1**	**1**
12	1	1	0	0	**0**	**0**
13	1	1	0	1	**0**	**0**
14	1	1	1	0	**0**	**0**
15	1	1	1	1	**0**	**0**

Nicht für den Prüfling bestimmt

<table>
<tr><th rowspan="2">Aufg.</th><th rowspan="2">erwartete Leistungen</th><th colspan="3">BE</th></tr>
<tr><th>I</th><th>II</th><th>III</th></tr>
<tr><td>2.3</td><td>bestimmen

$\boldsymbol{x} = (\bar{d} \wedge \bar{c} \wedge \bar{b} \wedge a) \vee (\bar{d} \wedge \bar{c} \wedge b \wedge \bar{a}) \vee (\bar{d} \wedge \bar{c} \wedge b \wedge a)$
$\vee (d \wedge \bar{c} \wedge \bar{b} \wedge \bar{a}) \vee (d \wedge \bar{c} \wedge \bar{b} \wedge a) \vee (d \wedge \bar{c} \wedge b \wedge \bar{a})$
$\vee (d \wedge \bar{c} \wedge b \wedge a)$</td><td>3</td><td>1</td><td></td></tr>
<tr><td>2.4</td><td>vereinfachen, nennen

x:
<table>
<tr><td></td><td></td><td colspan="2">a</td><td></td><td></td></tr>
<tr><td></td><td>0</td><td>1</td><td>0</td><td>0</td><td></td></tr>
<tr><td rowspan="2">b</td><td>1</td><td>1</td><td>0</td><td>0</td><td></td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td><td rowspan="2">d</td></tr>
<tr><td></td><td>1</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td></td><td></td><td></td><td colspan="2">c</td><td></td></tr>
</table>

$\boldsymbol{x} = (\bar{c} \wedge b) \vee (d \wedge \bar{c}) \vee (\bar{c} \wedge a)$

y:
<table>
<tr><td></td><td></td><td colspan="2">a</td><td></td><td></td></tr>
<tr><td></td><td>0</td><td>0</td><td>0</td><td>0</td><td></td></tr>
<tr><td rowspan="2">b</td><td>X</td><td>X</td><td>0</td><td>0</td><td></td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td><td rowspan="2">d</td></tr>
<tr><td></td><td>1</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td></td><td></td><td></td><td colspan="2">c</td><td></td></tr>
</table>

$\boldsymbol{y} = (d \wedge \bar{c})$

vereinfachen
nennen</td><td>
2</td><td>2</td><td>4</td></tr>
</table>

<table>
<tr><th rowspan="2">Aufg.</th><th rowspan="2">erwartete Leistungen</th><th colspan="3">BE</th></tr>
<tr><th>I</th><th>II</th><th>III</th></tr>
<tr><td>2.5.1</td><td>dokumentieren

Ein taktflankengesteuertes JK-Flipflop wird durch folgendes Schaltzeichen dargestellt:

Es besitzt drei Eingänge. Der Eingang C des JK-Flip-Flops ist der Takteingang, abgeleitet vom englischen clock (Takt). Hier sollte ein Rechtecksignal anliegen. Die beiden Eingänge J und K sind Steuereingänge. Das JK-Flip-Flop hat eine Steuerung auf der Taktflanke, also dem Übergang an C von 0 nach 1 oder umgekehrt von 1 nach 0, je nachdem, ob es auf die steigende oder auf die fallende Flanke reagieren soll. Die Ausgänge Q1 und Q2 werden in Abhängigkeit der Ansteuerung der Eingänge J und K gesteuert. Ein JK-Flip-Flop wechselt beim Anlegen eines Taktimpulses seinen Ausgangszustand, wenn an beiden Eingängen (J und K) H-Pegel anliegen. Dieses Verhalten wird als Toggeln (kippen) bezeichnet. Die Wahrheitstabelle eines taktflankengesteuerten JK-Flipflops lässt sich wie folgt darstellen:

<table>
<tr><th>C</th><th>K</th><th>J</th><th>Q_1</th><th>Q_2</th><th>Funktion</th></tr>
<tr><td>0 > 1</td><td>0</td><td>0</td><td>n</td><td>n</td><td>Speichern</td></tr>
<tr><td>0 > 1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>Setzen</td></tr>
<tr><td>0 > 1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>Rücksetzen</td></tr>
<tr><td>0 > 1</td><td>1</td><td>1</td><td>X</td><td>X</td><td>Toggeln</td></tr>
</table>
</td><td></td><td>4</td><td></td></tr>
<tr><td>2.5.2</td><td>entwerfen</td><td></td><td>1</td><td>3</td></tr>
<tr><td></td><td>Summe 28</td><td>10</td><td>11</td><td>7</td></tr>
</table>

Aufg.	**erwartete Leistungen**	**BE I**	**BE II**	**BE III**
3.1	erläutern MC Port 1 (out): Pin 3: Brennvorgang stoppen, Pin 2: Normaler Heizbetrieb, Pin 1: Brennvorgang starten Port 2 (in): Pin 7: Error, Pin 0-6: Thermostat Für den Intel-8086-Mikroprozessor muss jeweils ein Port als Eingang und ein Port als Ausgang definiert werden. Aus diesem Grund ist die vorgegebene Pinbelegung auf zwei Ports aufgeteilt worden: Port 10 in, Port 20 out.	4		
3.2	darstellen Pelletofen endlos – Raumtemperatur einlesen (Port 2, Pin 0-6) – Temp < 24° C ? J: –– Raumtemperatur einlesen (Port 2, Pin 0-6) –– Temp < 20° C ? J: ––– Brennvorgang starten beginnen (Port 1, Pin 1 = 1) ––– warten 2 Minuten aufrufen ––– Brennvorgang starten beenden (Port 1, Pin 1 = 0) ––– Errorbit einlesen (Port 1, Pin 7) ––– Kein Error? (Port 2, Pin 7 = 0) J: –––– solange Temp < 24° ––––– Raumtemperatur einlesen (Port 2, Pin 0-6) ––––– Normaler Heizbetrieb (Port 1, Pin 2 = 1) –––– Normaler Heizbetrieb beenden (Port 1, Pin 2 = 0) –––– Brennvorgang stoppen beginnen (Port 1, Pin 3 = 1) –––– warten 3 Minuten aufrufen –––– Brennvorgang stoppen beenden (Port 1, Pin 3 = 0) ––– Kein Error? N: –––– Alle Modi ausschalten (Port 1, Pin 1,2,3 = 0) –– Temp < 20° C ? N: nop – Temp < 24° C ? N: nop	4	6	

<table>
<tr><th rowspan="2">Aufg.</th><th rowspan="2">erwartete Leistungen</th><th colspan="3">BE</th></tr>
<tr><th>I</th><th>II</th><th>III</th></tr>
<tr><td>3.3</td><td>implementieren
org 100h
endlos:
IN AL, 2 ;Port 2 einlesen
CMP AL, 24d ;Thermostat < 24C
JB groesser20
JMP endlos

groesser20:
IN AL, 2 ;Port 2 einlesen
CMP AL, 20d ;Thermostat < 20C
JB brennvorgangStarten
JMP endlos

brennvorgangStarten:
MOV AL, 00000010b ;Pin 1 = 1 -> Brennvorgang starten an
OUT 1, AL
CALL wait2min
MOV AL, 00000000b ;Pin 1 = 0 -> Brennvorgang starten aus
OUT 1, AL

IN AL, 1 ;Ueberpruefung auf Error
SHR AL, 7
AND AL, 00000001b
CMP AL, 1
JNE normalerHeizbetrieb
JMP allesAus

allesAus:
MOV AL, 00000000b ;Pin 1,2,3 = 0 -> alle Modi aus
OUT 1, AL

normalerHeizbetrieb:
IN AL, 2 ;Port 2 einlesen
CMP AL, 24d ;so lange Temp <24 -> heizen
JAE brennvorgangStoppen ;wenn >24 Brennvorgang stoppen
MOV AL, 00000100b ;Pin 2 = 1 -> Normaler Heizbetrieb an
OUT 1, AL
JMP normalerHeizbetrieb

brennvorgangStoppen:
MOV AL, 00000000b ;Pin 2 = 1 -> Normaler Heizbetrieb aus
OUT 1, AL
MOV AL, 00001000b ;Pin 3 =1 -> Brennvorgang stoppen an
OUT 1, AL
CALL wait3min
MOV AL, 00000000b ;Pin 3 = 0 -> Brennvorgang stoppen aus
OUT 1, AL

JMP endlos
ret</td><td></td><td>3</td><td>7</td></tr>
</table>

Aufg.	erwartete Leistungen	BE		
		I	II	III
3.4	skizzieren, erklären Als Totzeit wird in der Regelungstechnik die Zeitspanne zwischen der Signaländerung am Systemeingang und der Signalantwort am Systemausgang einer Regelstrecke bezeichnet. Jede Änderung des Eingangssignals ruft eine um die Totzeit verzögerte Änderung des Ausgangssignals hervor. Die Totzeit ist in diesem Beispiel die Zeit, die verstreicht, bis die gewünschte Raumtemperatur nach dem Start des Ofens erreicht wurde und beträgt 35 Minuten. skizzieren erklären		 5 1	 1

<table>
<tr><th rowspan="2">Aufg.</th><th rowspan="2">erwartete Leistungen</th><th colspan="3">BE</th></tr>
<tr><th>I</th><th>II</th><th>III</th></tr>
<tr><td>3.5.1</td><td>erläutern, beurteilen

Ein pyroelektrischer Sensor, auch PIR-Sensor, (engl. Pyroelectric Infrared Sensor / Passive Infrared Sensor), ist ein Halbleitersensor, der zur Detektion von Temperaturänderungen dient. PIR-Sensoren beruhen auf der namensgebenden Pyroelektrizität, einer Eigenschaft einiger piezoelektrischer Halbleiterkristalle. Dabei führt eine Temperaturänderung ΔT zu einer messbaren Änderung der elektrischen Spannung. Die verwendeten pyroelektrischen Kristalle besitzen mit einer einzigen polaren Achse eine in der Kristallwelt seltene Asymmetrie. Dadurch ändert sich ihre Polarisation mit der Temperatur. Zur Nutzung des pyroelektrischen Effekts wird ein dünner pyroelektrischer Kristall senkrecht zur polaren Achse mit Elektroden beschichtet. Auf der oberen Elektrode des Kristalls ist eine absorbierende Schicht aufgebracht. Trifft Infrarotstrahlung auf diese Schicht, erwärmt sich das Pyroelektrikum und Oberflächenladungen entstehen. Die Ladungen sind jedoch sehr gering. Bevor sie sich durch den endlichen Innenwiderstand des Kristalls wieder ausgleichen können, wandeln extrem rausch- und leckstromarme Feldeffekttransistoren oder Operationsverstärker die Ladungen in eine Signalspannung um. Diese kann dann gemessen und ausgewertet werden.
Sollte die erläuterte Sensorart jedoch in einem Pelletofen eingesetzt werden, ist mit fehlerhaften Messergebnissen zu rechnen, da pyroelektrische Sensoren bereits auf kleinste Temperaturänderungen reagieren. Während des Brennvorgangs heizt sich nicht nur der Brennraum, sondern auch der Vorratsbehälter auf. Es könnte also passieren, dass die PIR-Sensoren durch die steigende Temperatur ausgelöst werden und damit Fehlsignale produzieren. Sie würden dann nicht das Absinken des Füllstands, sondern die Wärmeänderungen im Vorratsspeicher detektieren. Ein von der Firma HeatUp vorgeschlagener Einbau dieser Technologie ist also nicht sinnvoll.

erläutern
beurteilen</td><td>

3</td><td>

1</td><td>

3</td></tr>
</table>

Aufg.	**erwartete Leistungen**	**BE**		
		I	**II**	**III**
3.5.2	entwickeln Im Vorratsbehälter könnten zwei temperaturunabhängige Laser-Lichtschranken angebracht werden, Lichtschranke 1 in halber Höhe und Lichtschranke 2 kurz über dem Boden. Wenn beide Lichtschranken kein Signal liefern kann man davon ausgehen, dass der Vorratsbehälter gefüllt ist. Liefert zunächst Lichtschranke 1 ein Signal, aber Lichtschranke 2 nicht, sinkt der Füllstand. Man könnte in diesem Fall den Nutzer schon vorab informieren. Liefern beide Lichtschranken ein Signal, ist der Vorratsbehälter nahezu komplett entleert und der Nutzer muss benachrichtigt werden. Die Auswertung der Rohdaten der Lichtschranken könnte mithilfe eines Assemblers funktionieren, der dann bei entsprechender Datenlage weitere Aktionen anstößt. Über die Zeit könnte Sender oder Empfänger der Lichtschranke durch Staub (Abrieb der Holzpellets) verdrecken, was zu einer fehlerhaften Auswertung führen würde. Lichtschranke 1 Lichtschranke 2		2	4
	Summe 44	**11**	**18**	**15**

III Bewertung und Beurteilung

Die Bewertung und Beurteilung erfolgt unter Beachtung der nachfolgenden Vorgaben nach § 33 der Oberstufen- und Abiturverordnung (OAVO) in der jeweils geltenden Fassung. Bei der Bewertung und Beurteilung der sprachlichen Richtigkeit in der deutschen Sprache sind die Bestimmungen des § 9 Abs. 12 Satz 3 OAVO in Verbindung mit Anlage 9b anzuwenden.

Bei der Bewertung und Beurteilung der Übersetzungsleistung in den Fächern Latein und Altgriechisch sind die Bestimmungen des § 9 Abs. 14 OAVO in Verbindung mit Anlage 9c anzuwenden.

Der Fehlerindex ist nach Anlage 9b zu § 9 Abs. 12 OAVO zu berechnen. Für die Ermittlung der Punkte nach Anlage 9a zu § 9 Abs. 12 OAVO sowie Anlage 9c zu § 9 Abs. 14 OAVO wird jeweils der ganzzahlige nicht gerundete Prozentsatz bzw. Fehlerindex zugrunde gelegt.

Für die Bewertung in den modernen Fremdsprachen ist der „Erlass zur Bewertung und Beurteilung von schriftlichen Arbeiten in allen Grund- und Leistungskursen der neu beginnenden und fortgeführten modernen Fremdsprachen in der gymnasialen Oberstufe, dem beruflichen Gymnasium, dem Abendgymnasium und dem Hessenkolleg“ vom 7. August 2020 (ABl. S. 519) zugrunde zu legen. Demnach erfolgt die Bewertung und Beurteilung mit der Maßgabe, dass lediglich bei der Ermittlung des Prüfungsergebnisses (Note) aus Prüfungsteil 1 und 2 gerundet wird.

Darüber hinaus sind die Vorgaben der Erlasse „Hinweise zur Vorbereitung auf die schriftlichen Abiturprüfungen (Abiturerlass)“, „Hinweise zur Vorbereitung auf die schriftlichen Abiturprüfungen im beruflichen Gymnasium (fachrichtungs-/ schwerpunktbezogene Fächer) (Abiturerlass BG)“ und „Durchführungsbestimmungen zum Landesabitur“ in der für den Abiturjahrgang geltenden Fassung zu beachten.

Als Kriterien für die Bewertung und Beurteilung dienen unter Beachtung der Zielsetzung der gymnasialen Oberstufe nach § 1 Abs. 2 OAVO neben dem Inhaltlichen auch die in den Kerncurricula genannten überfachlichen Kompetenzen, insbesondere die Sprachkompetenz und Wissenschaftspropädeutik; dies zeigt sich u.a. in qualitativen Merkmalen wie Strukturierung, Differenziertheit, (fach-)sprachlicher Gestaltung und Schlüssigkeit der Argumentation.

Im Fach Technische Informatik besteht die Prüfungsleistung aus der Bearbeitung eines Vorschlags, wofür insgesamt maximal 100 BE vergeben werden können. Ein Prüfungsergebnis von **5 Punkten (ausreichend)** setzt voraus, dass mindestens 45% der zu vergebenden BE erreicht werden. Ein Prüfungsergebnis von **11 Punkten (gut)** setzt voraus, dass mindestens 75% der zu vergebenden BE erreicht werden.

Gewichtung der Aufgaben und Zuordnung der Bewertungseinheiten zu den Anforderungsbereichen

Aufgabe	Bewertungseinheiten in den Anforderungsbereichen			Summe
	AFB I	AFB II	AFB III	
1	9	11	8	**28**
2	10	11	7	**28**
3	11	18	15	**44**
Summe	**30**	**40**	**30**	**100**

Die auf die Anforderungsbereiche verteilten Bewertungseinheiten innerhalb der Aufgaben sind als Richtwerte zu verstehen.